

In The Claims:

Please amend claims 1, 10, 14 and 22-31 as indicated below.

1. (Currently amended) A method for managing a network, the method comprising:

a client generating a request for type information for an attribute or event pertaining to management of one or more managed network objects, wherein the request is expressed in an interface definition language, wherein the interface definition language is operable to define object interfaces across a plurality of platforms and across a plurality of programming languages, wherein each managed network object is a computer programming language object representing one or more devices on a network;

sending the request for type information to an object request broker;

a metadata gateway receiving the request for type information from the object request broker;

reading the type information from a metadata repository, wherein the type information is stored in a database format in the metadata repository;

translating the type information from the database format to the interface definition language;

the metadata gateway sending the translated type information to the object request broker; and

the client receiving the translated type information for the attribute or event through the object request broker, wherein the translated type information is expressed in the interface definition language.

2. (Original) The method of claim 1, wherein the translating the type information from the database format to the interface definition language comprises:

translating the type information from the database format to an abstract syntax notation; and

translating the type information from the abstract syntax notation to the interface definition language.

3. (Previously presented) The method of claim 2, wherein the abstract syntax notation is Abstract Syntax Notation One (ASN1).

4. (Original) The method of claim 2, wherein the translating the type information from the abstract syntax notation to the interface definition language comprises:

translating the type information from the abstract syntax notation to an object specification language; and

translating the type information from the object specification language to the interface definition language.

5. (Original) The method of claim 1, wherein the sending the request for type information to an object request broker, the metadata gateway receiving the request for type information from the object request broker, the metadata gateway sending the translated type information to the object request broker, and the client receiving the translated type information for the attribute or event through the object request broker are effected via an internet inter-object communication protocol.

6. (Previously presented) The method of claim 5, wherein the internet inter-object communication protocol comprises Internet Inter-Object Protocol (IIOP).

7. (Original) The method of claim 1, wherein the metadata gateway is implemented on a single server computer system.

8. (Previously presented) The method of claim 1, wherein the metadata gateway is distributed over a plurality of servers, wherein each of the plurality of servers presents a functionally identical view of the metadata gateway.

9. (Original) The method of claim 1, wherein the interface definition language is class-independent.

10. (Currently amended) A method for managing a network, the method comprising:

a client generating a request to encode type information for an object, attribute, or event pertaining to management of one or more managed network objects, wherein the request is expressed in an interface definition language, wherein the interface definition language is operable to define object interfaces across a plurality of platforms and across a plurality of programming languages, wherein each managed network object is a computer programming language object that represents one or more devices on a network;

sending the request to encode the type information to an object request broker;

a metadata gateway receiving the request to encode the type information from the object request broker;

translating the type information from the interface definition language to a database format; and

storing the type information in a metadata repository, wherein the type information is stored in a database format in the metadata repository.

11. (Original) The method of claim 10, wherein the translating the type information from the interface definition language to the database format comprises:

translating the type information from the interface definition language to an abstract syntax notation; and

translating the type information from the abstract syntax notation to the database format.

12. (Original) The method of claim 11, wherein the translating the type information from the interface definition language to the abstract syntax notation comprises:

translating the type information from the interface definition language to an object specification language; and

translating the type information from the object specification language to the abstract syntax notation.

13. (Original) The method of claim 10, wherein the sending the request to an object request broker and the metadata gateway receiving the request to encode the type information from the object request broker are effected via an internet inter-object communication protocol.

14. (Currently amended) A network management system comprising:

a metadata repository, wherein the metadata repository comprises metadata concerning object classes for a plurality of managed objects, wherein the metadata comprises information expressed in a database format, and wherein the managed objects are computer programming language objects corresponding to managed devices on a network; and

a metadata gateway which is communicatively coupled to the metadata repository and to an object request broker, wherein the metadata gateway is operable to send and receive the metadata from the database, wherein the metadata gateway provides translation of the metadata to and from the database format and an interface definition language, wherein the interface definition language is operable to define object interfaces across a plurality of platforms and across a plurality of programming languages.

15. (Original) The network management system of claim 14, wherein the managed devices comprise a telephone system.

16. (Original) The network management system of claim 14, wherein the managed devices comprise a network switch.

17. (Original) The network management system of claim 14, wherein the metadata gateway further comprises:

a library of data types expressed in an abstract syntax notation, wherein the abstract syntax notation comprises a metadata notation language;

a plurality of object types, wherein each object type comprises one or more of the data types from the library of data types; and

an interface to the plurality of object types, wherein the interface is operable to provide one or more clients with access to the metadata as expressed in the interface definition language.

18. (Original) The network management system of claim 17, wherein the interface to the plurality of object types is a programming-language-independent and platform-independent interface.

19. (Original) The network management system of claim 17, wherein the plurality of object types comprise CORBA objects.

20. (Original) The network management system of claim 14, wherein the object request broker is configurable to be accessed by a plurality of network management clients to obtain the metadata as expressed in the generic interface.

21. (Original) The network management system of claim 14, wherein the object request broker comprises a CORBA ORB.

22. (Currently amended) A tangible, computer-readable storage ~~accessible~~ medium comprising program instructions, wherein the program instructions are computer-executable to implement:

a metadata gateway receiving a request for type information from an object request broker, wherein the type information pertains to management of one or more managed network objects, wherein each managed network object is a computer programming language object that represents one or more devices on a network;

reading the type information from a metadata repository, wherein the type information is stored in a database format in the metadata repository;

translating the type information from the database format to an interface definition language; and

the metadata gateway sending the translated type information to the object request broker.

23. (Currently amended) The computer-readable storage accessible medium of claim 22, wherein in translating the type information from the database format to the interface definition language, the program instructions are further computer-executable to implement:

translating the type information from the database format to an abstract syntax notation; and

translating the type information from the abstract syntax notation to the interface definition language.

24. (Currently amended) The computer-readable storage accessible medium of claim 23, wherein the abstract syntax notation is Abstract Syntax Notation One (ASN1).

25. (Currently amended) The computer-readable storage accessible medium of claim 22, wherein in translating the type information from the abstract syntax notation to the interface definition language, the program instructions are further computer-executable to implement:

translating the type information from the abstract syntax notation to an object specification language; and

translating the type information from the object specification language to the interface definition language.

26. (Previously presented) The computer accessible medium of claim 22, wherein the interface definition language is class-independent.

27. (Currently amended) A tangible, computer-readable storage ~~accessible~~ medium comprising program instructions which are computer-executable to implement:

a metadata gateway receiving a request to encode type information from an object request broker, wherein the type information pertains to management of one or more managed network objects;

translating the type information from an interface definition language to a database format; and

storing the type information in a metadata repository, wherein the type information is stored in a database format in the metadata repository.

28. (Currently amended) The computer-readable storage ~~accessible~~ medium of claim 27, wherein in translating the type information from the interface definition language to the database format, the program instructions are further computer-executable to implement:

translating the type information from the interface definition language to an abstract syntax notation; and

translating the type information from the abstract syntax notation to the database format.

29. (Currently amended) The computer-readable storage ~~accessible~~ medium of claim 28, wherein the abstract syntax notation is Abstract Syntax Notation One (ASN1).

30. (Currently amended) The computer-readable storage ~~accessible~~ medium of

claim 27, wherein in translating the type information from the interface definition language to the abstract syntax notation, the program instructions are further computer-executable to implement:

translating the type information from the interface definition language to an object specification language; and

translating the type information from the object specification language to the abstract syntax notation.

31. (Currently amended) The computer-readable storage ~~accessible~~ medium of claim 27, wherein the interface definition language is class-independent.